A Token-based Access Control Mechanism for Automated Capture and Access Systems in Ubiquitous Computing[1]

Giovanni Iachello and Gregory D. Abowd

College of Computing and GVU Center
Georgia Institute of Technology
801 Atlantic Dr.
30332-0280 Atlanta, GA, USA
{giac, abowd}@cc.gatech.edu

Abstract. We discuss the problems related to access control in automated capture and access systems, which capture, store and retrieve information gathered through sensors in physical environments. We discuss several unique requirements that set capture and access apart from traditional information processing systems, and that make existing access control approaches such as role-based access control (RBAC) and digital rights management (DRM) unsuitable for this domain. Drawing from access control theory research, we devise an access control system that satisfies these requirements. Further, we describe its implementation within an existing capture and access system, and discuss emergent issues relating to retention time, rights management and information sharing. We argue that some traditional security requirements might not in fact be appropriate when applied to environmental captured information, due to the perceptual and social characteristics of such data. Finally, we provide an example of how this access control architecture might fit in a capture and access system composed of mobile devices.

Introduction and Related Work

Automated capture and access (CA[2]) systems have emerged from the need for many different ubiquitous computing (ubicomp) applications to support the collection and retrieval of information from the physical world, such as audio and video feeds, the location of individuals and objects, or other kinds of environmental information (*e.g.*, temperature, concentration of airborne chemicals). Capture and access systems have been used to build, among others, recording applications for meetings [15] and classrooms [1], personal memory aid tools [12, 21, 8], and therapeutic support systems [23]. Design and architectural considerations are leading to the consolidation of collection and retrieval functions in generic infrastructural systems, similarly to data networking services, available to multiple applications. Due to their instrumental

[1] This article is available at www.gvu.gatech.edu as Georgia Institute of Technology GVU Center Technical Report GIT-GVU-05-06.

[2] Not to be confused with the same acronym for Certification Authority.

nature, CA technologies are on their way to becoming one of the upcoming large information infrastructures.

Securing collected data is an obvious requirement for CA systems. Privacy and security have represented concerns in the wider ubicomp community from the beginning of the '90s [22, 2]. Early work concentrated mostly on social acceptability and legal issues, especially in relation to privacy, and access control has not been generally addressed due to the priority given to functional issues. Recently however, researchers has started analyzing access control problems in ubiquitous computing systems from various perspectives. For example, Sampemane *et al*. have shown how to employ generalized role-based access control (RBAC) to secure ubiquitous information services [20] (*e.g.*, for annexing information or computing services overlaid with physical environments), and Covington has used environmental sensed data as inputs to RBAC evaluation functions [7]. As opposed to the two mentioned approaches, we are interested in providing access control for the information collected in physical environments — not for ubiquitous services — in a way appropriate to the perceptual characteristics of such data.

We propose to associate a set of tokens to each environmental data item stored by the CA system; tokens are also distributed to the users who need to access the information. The user must subsequently show the token to the system in order to gain access to that data item. This approach is similar to the key-lock pair access control techniques first used in early multiprocessing systems and documented by Graham and Denning [10]. In that article, Graham and Denning also describe capability lists, in which each user is associated with a set of access capabilities on data and access control lists (or ACL), in which objects are associated with access permissions. However, our system differs from their key-lock approach in that users do not need to be identified ahead of time and tokens are automatically associated at the time of capture with incoming streaming data. Token-based access control also differs from capability lists in that the user, not the system, keeps track of tokens. Further, token-based access control differs from both capability lists and ACL in that it does not require to identify users to evaluate the access control functions (*i.e.*, to grant or deny access).

Tokens can be thought of as "memory handles" which allow access only to the segments of information that a user chooses, or is allowed, to "remember," as proven by retaining the token. The effectiveness of this access control scheme depends on how the captured data items are annotated with location and temporal attributes. Moreover, tokens have rather weak security properties in some respects. We argue below, however, that considering automated capture and access in its broader context, these weaknesses might be the result of the intrinsic nature of people's understanding of everyday experience, and that it might be impossible to control such data exchanges by using any access control technique.

Below, we present the requirements analysis of an access control system for CA applications and describe how we implemented it within an existing infrastructure system, the InCA toolkit [19]. We do expose how the structure of InCA has influenced part of the design of the access control system, and which aspects are independent of InCA. We further develop some comments on how access control

could be used in a variety of environmental capture applications, concentrating specifically on the case in which personal, portable terminals can be used as brokers between users and system components.

Requirements

The characteristics of CA systems make it hard to support access control by using traditional RBAC approaches, or discretionary access control (DAC, used *e.g.* for file access control in UNIX), mandatory access control (MAC, used *e.g.* in classified systems), or digital rights management (DRM) techniques, used for published multimedia data, due to a variety of reasons.

First, CA systems used in ubicomp applications cannot assume that users are known before a certain recording takes place, and access to the information should be granted to anonymous users, or users identified by pseudonyms. Second, the architecture should allow sufficient flexibility in the definition of access policies to support the vast range of usage settings and applications supported by CA systems. Third, the access control system should support the unique properties of environmental information, including its relationship with the geographical and temporal locations of collection. Finally, the system's implementation should be lightweight.

Below, we discuss these requirements individually. This analysis is not intended to be exhaustive; rather, our intention is to highlight some of the unique characteristics brought on by environmental CA technology, to point out architectural implications, and to suggest some problematic areas in need of further investigation.

Undefined user set

The infrastructure should not assume that subjects can be identified, either as related to human users or as specific applications. This requirement can be broken down as follows.

First, the presence of a specific user during automated capture often cannot be predicted; unplanned meetings in informal organizations are an example (*e.g.*, friends, households, or even open formal organizations such as schools). Thus, access control methods which require prior registration with a centralized entity would not fit well with such applications. This includes traditional techniques like DAC and MAC, but also schemes specifically developed for ubicomp applications, such as Sampemane's [20].

Second, diverse accessing entities should be supported by the access control method, including users and software processes, because ubicomp systems are composed of heterogeneous devices. This implies that the access control system should not make overly specific assumptions on the capabilities of the accessing entity, such as the ability to perform cryptographic operations.

Third, it should not be necessary to identify users of the information to control access to stored data. While anonymous and pseudonymous access control schemes are documented in the literature, they might not be appropriate for the present application. Specifically, unlinkable anonymous access control frameworks such as Idemix [4] require a trusted third party (TTP) to issue digitally signed access credentials, that are then blindly verified by the access control system. Given that a single semantic unit of information might include references to several hundreds or thousands of individual data items this would tax the capture and access with a high signature verification overhead if one credential were to be used for each accessed data item. On the other hand, if Idemix credentials were to be issued containing all access control permissions for a certain user, we still would have the problem of verifying access to individual items. Thus, we have chosen to employ a simpler technique that does not involve cryptography. However, it could be possible to integrate Idemix with our technique, in order to provide strong anonymity and limit the sharing of access credentials among different subjects.

DRM techniques are used for access control to multimedia information, and could even be extended to "weak anonymous"[3] access control. However, DRM systems (such as those in current commercial use or those conformi g to specifications like [6]) tend to be very resource intensive, involving TTP, digital signature schemes and trusted implementations. Moreover, these systems are biased towards supporting relatively few publishers and many information consumers. Finally, information is published relatively rarely compared with the amount of times the information is accessed. In that perspective, concentrating the computational effort during the publishing phase results in a practical architecture.

n

DRM systems might not be a viable option for ubiquitous CA because many assumptions are different; first, in CA environments, each user is potentially both a consumer and a publisher of information; second, in many CA applications the ratio between the occurrences of data generation and data access is much closer to 1; third, the data requiring access control are most typically continuous streams, as opposed to the discreet data model assumed in DRM systems (the copyrighted, published "piece of intellectual work"); finally, and most important, in many situations information generated by CA systems cannot be attributed to any single owner.

Flexibility in the definition of access policies

In ubicomp applications, vastly diverging usage needs might require unorthodox access control policies. Here, our intent is not that of defining any specific policy. Our objective is to describe a basic access control method that could, with the help of external rules, generically support many different policies required in automated CA.

[3] Weak anonymity is defined in opposition to cryptographic, or "strong," anonymity techniques [5]. Weak anonymity is not resistant against a concerted attack brought on by linking different access instances, but can be effectively used in real-world applications when it raises the cost of attack beyond what would be acceptable to an attacker.

To better understand the different usage context and policies for automated CA, consider the following three examples:

1. In a classroom setting, the CA system could be used to store video and audio footage of lectures, which should be accessible by all students registered to the course (even if they are not present at one specific lecture) and by the instructors.
2. Captured corporate meetings should be accessible by all individuals invited to the meeting; such a policy might not be trivial to implement: in many organizations, long meetings are composed of different sections, some of which might be confidential, and some individuals might be asked to temporarily leave the room. These persons should be allowed access to the portions of the meeting they attended and denied it to the remaining parts.
3. A third example might involve capture in a private environment, where the owner of a dwelling might want to have access to captured information regardless of his or her presence at the time of capture.

These examples show that an access control system for CA infrastructure should support at least content-based, location- and time-based, and identity-based policies:

- *Content-based* policies should support access control based on the object of the environmental recording (*e.g.*, all instances of lectures associated with a specific class). It should be noted that this policy is not necessarily connected with the location of capture, or the specific capture devices. In the classroom example, the class might gather in a different room or go on a field trip, so the access control should be independent of the location or the identity of the capture device.
- *Location and time-based* access policies should grant to a user access to environmental information gathered when he or she was present, for example to support the above corporate meetings, or other "memory aid" applications which provide an archive of what happened during the presence of the individual for later use.
- *Identity-based* policies are similar to current access control schemes based on true identities or pseudonyms, and should be supported as well. These traditional access control systems grant access based on the intrinsic identity of an individual or to his or her belonging to an organizational group or role. Given the unpredictable usage contexts of CA applications, these types of policies are not to be considered exhaustive: the system should be flexible enough to allow for future expansion.

Support the unique properties of environmental capture information

Environmental captured information is, by its very nature, subject to unique interpretation and understanding by users, based on the *context* of capture (*e.g.*, time, location, people present, activity involved, *etc.*). Ubicomp researchers have attempted to characterize these concepts by drawing from sociological and cognitive theories, and have been especially influenced by phenomenology (see for example [11]). The present work is inspired by the observations proposed in that context.

If the access control policy requires the physical presence of the principal (*e.g.*, a user or his/her agent, or a process) at the time and location of capture to be granted later access, both the presence sensor and the access control policy must be sufficiently fine-grained to support "secure" determination for access control. For example, if access to recorded information is conditional on a person's presence in a room, and a certain amount of time is sufficient to leave or enter that location (*e.g.*, five seconds), access rights should refer to capture units of appropriate length. In this case, access control information should be associated to data segments as small as those containing five seconds of audio recording or video footage. A similar characterization of access to environmental information (*i.e.*, granting access based on experiential determinations) has been recently proposed by Duan and Canny [9], who use the term "Data Discretion Principle" to describe it. Their proposal involves however the use of key sharing and encryption to enforce the principle. We are interested in understanding whether more lightweight techniques can be used that do not require the management burden associated with encryption technologies. While encryption could be used in our scheme for added protection (*e.g.*, encrypting data in transit or in storage), we do not want our access control scheme to rely on it.

This objective can be achieved in various ways. Access control policies associated to a long recording could specify subsets of the recording to which a principal is allowed access. This would however require the infrastructure to be able to interpret the data format of the stored information, to extract timing, location and other parameters relevant to the access control determination. An alternative would be requiring capture applications to segment environmental information in perceptually coherent chunks and store these chunks, instead of the complete uninterrupted recording. Access control can then be implemented on the chunks instead of on the whole recording, thus eliminating the need of interpreting the content of the recording itself.

Lightweight implementation

CA systems must support high bandwidth transactions, including multiple video and audio streams, as well as other kinds of potentially voluminous data flows. Moreover, in many situations, the stored information might be accessed only a few times in its lifetime within the system. Thus, expensive publishing schemes such as anonymous credentials or DRM might be inappropriate.

System Design

Given the requirements analysis outlined above, we decided not to base CA access control on identifiable users, but on "knowledge tokens." As long as the accessing entity possesses the required token, it can access the stored information. We implemented the access control system as an integral part of InCA, an experimental

infrastructure for environmental capture and access that has been used in several research projects.

We chose InCA as a test bed because it has been used in a variety of systems, over a lengthy timeframe, and it has been also used in actual "production" settings (e.g., class-room recording) [19, 14]. Furthermore, InCA code is readily available and modular, which greatly reduced the learning curve required in modifying some of its main components. As we explain below, however, the present work is not dependent on InCA itself: the concept of access tokens can be applied to other environmental capture technologies as long as relevant assumptions are met.

Architectural considerations suggest that access control should be implemented at the same infrastructural level as the CA system. Access control support positioned lower in the infrastructural stack (*e.g.*, operating system or database) does not provide sufficient flexibility for the type of information stored in CA systems, and higher level support might not guarantee effective enforcement. For this reason, the present access control mechanism was implemented as a component of an existing CA middleware solution.

Description of InCA

InCA is a system that provides abstractions for modeling capture, storage, delivery and query-based access to multi-media environmental information, including video, audio, digital ink strokes and text. The infrastructure is network-based, and is composed of a collection of independently running modules connected by a TCP network. InCA is a modular system, written in Java, composed of one core broker node (the InCA daemon), which runs at a well-known network address, and an arbitrary number of modules providing capture, access, control, observation, storage and transduction functions. (See Figure 1)

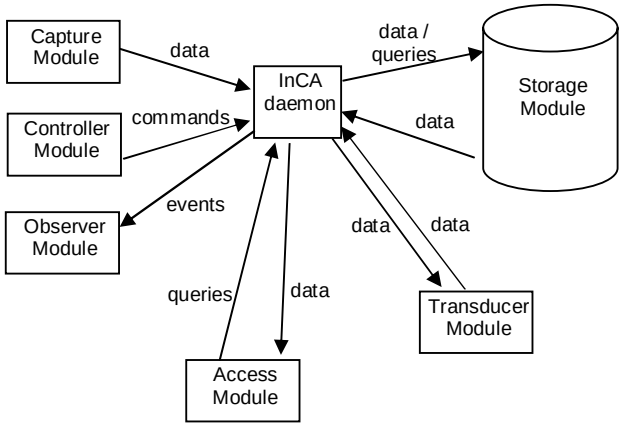

Fig. 1. InCA Architecture.

Data are collected by capture modules, and are tagged with `(key, value)` attribute pairs specifying the context of the capture, including location, time, people present, *e.g.* `("Location", "Room 330")`. Data transit through the broker to a storage module where they are archived, either in core memory, or in a database. Typically, continuous data is stored in the infrastructure as short segments, to improve access precision, since data can be accessed only at the segment level, through its attributes.

A query is made by an access module, usually part of the application making use of the data. Queries are evaluation functions of keys and values conjoined by logical operators, and are based on the attributes mentioned above (*e.g. "all data with date ≥ Tuesday, Sept. 7, 2004 and room = Room 315 and building = Hall X"*); usually these queries are generated programmatically, and not made directly by the user. The content of data items themselves is opaque to the infrastructure; thus, it is not possible to search on the contents of the data, but only on the associated attributes. Access modules request data from the InCA daemon, which contacts all registered storage modules for relevant data. All storage modules that hold relevant data return them to the daemon, which forwards them to the requesting access module. In addition, access modules can receive copies of the data at the time of capture if they have subscribed a prior query with the daemon.

Controller and observer modules are used to control the functioning of the infrastructure (*e.g.*, for activating or deactivating capture) and for observing its status (*e.g.*, when a module joins or disconnects), but do not present access control issues of interest. Finally, transducer modules are used to translate data between formats, at the time of capture or once it is stored, *e.g.*, translating raw audio footage into its transcript. Transducer modules can be modeled as composed of one access module and one capture module; nevertheless, they do present more complex access control issues, especially relating to token propagation between source and processed data. We do not however consider them here for the sake of brevity.

The access control system

Each data item stored in the CA system is associated with a (potentially different) set of secret tokens during the storage phase. The tokens are stored along with the environmental data and their attributes. Principals who are granted access to the stored data are provided with a copy of the relevant tokens.

The central role of the InCA daemon makes it the prime candidate for implementing access control. In a decentralized system, access control would have to be implemented within storage modules. In keeping with the existing design philosophy, we implemented only the bare minimum enforcement code in the infrastructure itself; policy definitions are allocated in external access control manager modules. This module manages the relationship between the infrastructure and users and applications, translating high level access policies into control metadata amenable to the daemon.

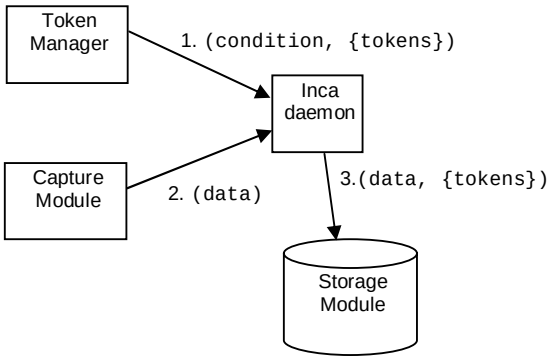

Fig. 2. Capture phase.

In InCA all transactions occur through the use of queries on the context attributes associated with the data. Some context attributes are generated automatically (*e.g.*, timestamp), but in general the capture application enriches the environmental data with specific and semantically dense information. The central role of queries in this system suggests to use them as the basis for the access control mechanism as well. Consequently, the structure and value of these attributes is integral to access control policy definition.

The system provides flexibility in how the tokens can be associated with data items during storage. An external token manager is used for this purpose. (See Figure 2) Such a design meets the first two requirements above (unknown user set and flexibility), because the access policy can be defined externally to the infrastructure. Individually tagging data items allows maximum flexibility in the determination of whom to grant access to them.

The token manager registers a set of tuples, composed of a condition and a set of tokens `(condition, {tokens})`, with the infrastructure; after such registration phase, all subsequent data bound for storage and which satisfy any of the conditions will be tagged with the associated tokens (*e.g.*: tag all data for which `"Location"` equals `"Room 330"` with tokens `x` and `y`). Conditions, similarly to queries, are evaluation functions on the attributes of the data. More than one condition may match a certain data item: in that case, all pertinent tokens will be associated with the data. The registered tuples can be removed or updated by the token manager when necessary. Changes to the registered tuples apply to all information stored subsequently, until the next change.

We make tokens opaque to the CA system to allow for generic implementations (the only operations required on the tokens are comparisons and conversions to and from strings for storage purposes). In a first version, tokens are pseudorandom 128 bit strings. Tokens are not generated within the capture module or the infrastructure, but in the token manager, in order to separate concerns between capture, access control and policy definition.

When a query for data is made (see Figure 3), the system examines the tokens associated with the data matching the query (step 5). If no tokens are associated with the stored data, access is granted (this maintains compatibility with existing applications, and is consistent with the view of tokens as conditions on future access to the data). If any tokens are present with the stored data, then only those data items are returned to the requesting application for which the application has demonstrated to possess at least one of the stored tokens. If the application did not provide satisfactory tokens, the query returns the null set (just as if the requested data did not exist). Principals indicate their possession of tokens by loading a set of known tokens into the system prior to making a query for environmental data (step 1). Figure 4 shows this same concept, comparing a traditional DAC access control matrix with token-based access control [10].

The potentially large number of tokens accumulated by CA applications presents the classic problems of large capability lists [10]. For this reason, principals should keep track of which tokens are relevant for a specific query (or group of queries), for example by grouping them by time, location or type of application, and load only tokens which are likely to be relevant into the system. Similarly, on the server side, data items may have a large number of associated tokens (this is similar to the classic problem of large access control lists). While the underlying database system provides fast search tools, this could nevertheless represent a bottleneck and would have to be addressed in a robust deployment (*e.g.*, by organizing tokens in binary search structures).

Above, we mentioned two alternatives for access control granularity on sensed data, namely, associating access control to portions of uninterrupted data streams, or to coherent chunks (in whatever dimension makes sense for the access control determination, *e.g.*, time and space). In our implementation, we adopted the latter solution because in InCA single data items are opaque once they enter the infrastructure, and the data model favors the segmentation of data streams in chunks (to increase access efficiency and precision). This however need not be: our token-based system could be adapted to CA systems handling continuous media as long as the following prerequisites are satisfied: 1) it is possible to address segments that are perceptually distinct for access control purposes (*e.g.*, 1 minute intervals of a video recording), and 2) it is possible to evaluate an access control function on each segment of interest. In this situation, tokens would not refer to physically distinct data objects, but to parts of data streams, which would need to be extracted from the data

A	Object X	Object Y
User 1	Read, Write	-, Write
User 2	Read, -	-,-

B	Object X	Object Y
	Token1, Token2	Token3

Fig. 3. Left, a typical access control matrix A for controlling read/write on objects; read access is granted to if `'Read' ∈ A[User, Object]`. Right, an access matrix B for token-based access control scheme; access is granted if `Token ∈ B[Object]`. Note that there are no a users, and that this scheme only controls read ccess (write access could be easily added, by associating an access mode to tokens).

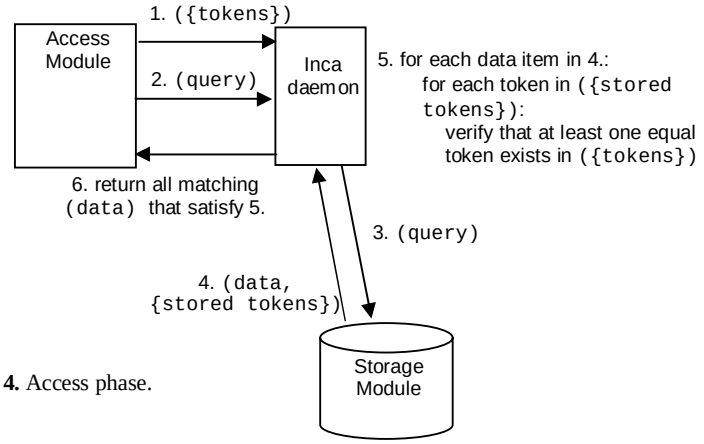

Fig. 4. Access phase.

streams by the infrastructure before returning them to the requesting entity. In any case, the fundamental property of tokens is that they represent an access permission to a specific portion of environmental data.

Finally, this system satisfies also the fourth requirement stated above, namely that the access control method be sufficiently lightweight. The only requirement for accessing stored information is the ability to store tokens, along with some related contextual information. The quantity of these tokens can be relatively large, but proxy services or functions can be used to reduce the amount of storage space required (*e.g.*, computing tokens based on a user secret (or password) and the query to the data, or storing the tokens within a broker service).

The external token manager supports a variety of setups in how the tokens can be generated and distributed, as shown by the following two examples.

Example 1. The token manager creates a new (random) token for accessing the information that is currently being stored each 10 seconds. These tokens are distributed to all principals present in the environment where the capture takes place, via a short-range wireless broadcast or line-of-sight communication mechanism. The tokens are collected by an agent on behalf of the principal (*e.g.*, an application running on an individual's cell phone or PDA). The users then use the collected tokens to access the stored information (*e.g.*, the recording of a meeting) at a later time. If a user leaves the environment, he or she will not be able to access the information, unless another knowledgeable individual provides him or her with the relevant tokens.

Example 2. The tokens are generated individually for each principal (thus, allowing to distinguish different users, but not necessarily compromising weak anonymity), based on a user-defined password or on other kinds of information (*e.g.*, biometric).

For example, present individuals may register with the token manager by typing an individual secret on a publicly available terminal in the conference room. The secret is then converted to a token using a well-known method, (*e.g.*, a secure hash function). Principals will need to remember such secret for further access to the data. Clearly, knowledge-based and identity-based token generation algorithms can be combined to support more complex usage scenarios.

In both examples, user identity authentication can be achieved by associating tokens to strong identity credentials (*i.e.*, generating them based on an individual secret which is expensive to disclose). Furthermore, a broker system, accessible through user authentication, could store the tokens associated with a specific individual, thus supporting a traditional access control policy.

Discussion

If viewed from a traditional security perspective, it is straightforward to identify weaknesses regarding the access control system described above, the most important being 1) that the described system does not allow to modify access permissions once they have been granted, 2) that it does not restrict principals from trading tokens in order to gain unintended access to information, and 3) that it only controls access (read) rights and not modification or deletion of data. We claim that the usage context and characteristics of CA require to step back and reconsider whether it makes sense at all to consider these three points as security issues. Although a variety of other issues can be pinpointed (vulnerability to replay attacks, to exhaustive search attacks, to impersonation attacks by modules, to eavesdropping communications among modules, etc.) we will concentrate on these three because they are most salient to the properties of ubicomp applications.

Modification of access rights

The current implementation does not allow to change access rights once they have been granted. This property is somewhat symmetric to what happens in the Chinese Wall policy [3], in which access rights cannot be changed once the user performs certain actions. Although it is indeed possible to extend the proposed access control system with an access rights manager that implements such requirement, by removing specific access tokens related to a principal (if such relationship is known), we would like to make here a different point, namely, that the perceptual qualities of physically sensed information are *phenomenologically* incompatible with rights revocation. In fact, if we view physical environmental data collection as a memory augmentation system, requiring to remit access rights might be akin to requiring to forcefully *forget* the memory of a certain experience or event.

Currently legal arrangements do exist, which require such performances by individuals and organizations (*e.g.*, non-disclosure agreements, retention limitations

in personal data protection regulation, *etc.*). The classic example is that of an individual leaving a certain organization and surrendering the access rights to information which is property of the organization. Their effectiveness is, however, based on their underpinning social framework, and only partially on technological enforcement. In many "open" applications, on the contrary, the ability to preserve access rights to environmental information for all involved stakeholders would avoid the social risk of a "privatization of experience". Thus, not providing the ability to revoke access rights could as well be justified as a deliberate design choice. We would like to stress that we are not advocating the validity of one particular view; our intent is just to highlight how social and cognitive considerations about the collection of environmental information can have profound implications on the design of access control systems.

Token trading

The second main weakness of the proposed access control infrastructure is that lacking an externally defined policy (*e.g.*, by implementing a custom token manager and broker), the system does not discourage principals to trade tokens in order to access other's data. While password trading happens also in traditional systems, it is curbed by the significant trust and responsibility risks bound to the fact that passwords grant access to the entire data and identity of the users. Tokens only relate to a limited set of data items and are thus less risky to disclose. Nevertheless, an argument grounded on the properties of captured environmental information, as perceived by individuals, can explain how such deficiency might be acceptable. If environmental capture is likened to a memory augmentation technology, trading tokens might be viewed as a technologically empowered way of disclosing to a third party what happened at a certain place and time. The disclosing individual might not have any control on how the third party may subsequently use that knowledge.

Clearly, the richness of the recorded information (*e.g.*, video/audio footage) could grant it an entirely different legal and social status than what usually associated to verbally transmitted recollections ("hearsay"). This difference is one of the most interesting social implications of ubicomp technologies [17]. The unique properties of rich environmental data transcend access control issues; in fact, independently of the particular mechanism adopted, access control alone is insufficient for protecting individuals from unwarranted disclosures, lacking a social and organizational underlying framework. This leads us to uncharted legal waters, pertaining to how individuals expectations of privacy are defined and modified in environments where high-fidelity recordings of actions and utterances take place. What we would like to stress here is that independently of the access control method, the interplay of environmental recordings with social customs does unsettle established privacy balances. Whatever the access control policy might be, it needs to be complemented with appropriate social norms, rules and practices.

An additional interesting property of the system, which can be also justified using phenomenological arguments, is that if nobody (be it a principal, or the token manager) collects and saves the tokens used during capture, that information will not

be accessible to any principal. Such a situation might arise because nobody is interested in the collected information, but also if the tokens have been lost or destroyed. Thus, in such a CA environment, users might adopt a conservative approach and collect all generated tokens, and decide later whether to keep them or not. This is possible with our access control system. For example, if a CA system is employed in security applications, such as surveillance, a token manager (which effectively implements access policies) could collect and securely store all tokens generated in the capture activity for later review, or always register a "master token" to be used later when particular circumstances (*e.g.*, an outstanding warrant) require to access stored data.

Control on retention and modification

Data retention policies play an important role in CA systems, as CA systems can be used to collect large amounts of personal information. Data retention time is a concept introduced by data protection legislation as a fundamental privacy-enhancing principle [16]. It refers to the length of time that personal information is kept stored before being destroyed. Such duration is connected to the intended uses of the information, which must be spelled out in advance of collection: in general personal information may be stored for no longer than is necessary to carry out the declared purpose. However, in CA systems, different principals with an interest in the data might have diverging opinions on the appropriate uses, and thus on the retention time for stored data.

Even if such a compromise could be met as a precondition to collecting the data, a principal might have a direct interest in amending or destroying previously stored environmental information. So data retention interacts with access control, because access rights to environmental information may also include the rights to amend or destroy the information. For example, one individual might want to eliminate footage that could be interpreted as socially inappropriate, or potential ground for litigation. In such case, a removal function in InCA could verify that the principal requesting the deletion or substitution of the data does possess the appropriate rights, similarly to what happens before granting access for retrieving data from storage. This would require to augment stored tokens with access rights such as "delete," "write" and "append".

Such a scenario, however, raises the question of how to reconcile divergent requests such as a principal asking to delete a certain portion of video footage that are of interest to others. Arguably, the quality of the "access rights" described in this section refer to emergent stakeholder rights, rather than to rights granted by authority, but it is not clear what rights should be granted to principals within CA spaces. Arbitration and adjudication issues have hitherto been only scratched by the ubicomp research community [13], and the ethical implications of amending or deleting portions of environmental recordings with multiple stakeholders are still unsolved.

For these reasons we have decided only to concentrate on access rights, and not on editing or removal.[4]

Summary

Our research has led us to the conclusion that rather than trying to enforce strong security in ubicomp systems, it is more productive to focus on misuse prevention, damage reduction and redress, by employing a mix of technical and social measures. The system proposed here should be assessed in this perspective: while it does not afford the level of security of a strictly administered operating system in a disciplined organization, when coupled with sound policies (implemented technically in token managers and within the organization that used the CA system) it can provide sufficient security for many CA applications. The system based on secret tokens is both *lightweight*, and *expansive*. It is lightweight because it does not impose strong requirements on the accessing entities. It is expansive because it tends to favor the diffusion of information rather than restricting it, and relies on social barriers to curb such diffusion to a socially acceptable rate, by requiring explicit trading of tokens, similarly to how we communicate our everyday experience. In this, it is similar to Povey's optimistic security [18], in which access rights are granted by default and abuse is controlled and if necessary remedied through subsequent audit. However, unlike that approach, our does not rely on the audit support (including principal identification) necessary to prosecute abuse, given the loose administrative environment of ubicomp applications which is ill-suited to enforcing centralized and secure audit functions.

Sample Application

We finally discuss an example of how the proposed access control system could be used to define complex policies in an environment populated by mobile CA devices. The purpose of this is both to detail the working environment of CA systems and to show how the lightweight system described here could be applied to such an architectural setup.

Mobile CA is of interest because mobile devices are becoming increasingly powerful, both in terms of capture and storage capabilities. Their increased power allows them to work within CA systems, which had been previously restricted to high-performance, dedicated, fixed computing systems. The personal nature of these devices, and the fact that people are growing increasingly accustomed to carrying them constantly (consider cell phones) make them excellent candidates as remote sensors, personal wallets, and for implementing agent-like functions like collecting

[4] As a more general comment, the role of the data subject in data protection legislation might need to be adapted to these new technologies, which expose situations where multiple individuals can claim privacy rights on information.

access tokens. Mobile devices incorporating all CA functions have been deployed [12] in a restricted configuration. Although in the cited case the CA application is integrated onto a single software module and does not support network communication, advanced systems will soon allow access to other devices both remote and present in the environment.

Architectural considerations suggest to analyze a mobile capture environment in terms of the location of relevant system components, namely the mobile device and a fixed infrastructure. We discuss the case in which the mobile device captures, stores and provides access to captured information (see below for comments on other architectures). The personal nature of mobile devices biases the access control policy through its intrinsic affordances (namely, the fact that the device is in continuous physical control of its owner). However, introducing a token distribution system can enable several useful policies.

Access tokens transmitted through out-of-band means (*e.g.*, by infrared communication) could support access policies which are perceptually understandable by their users. Tokens distributed in such a way may allow only devices present in the same environment to access the data stored by the mobile CA system, similarly to making a copy of minutes for all participants to a meeting. In this configuration, the capture module and token manager reside on the user's device. Other devices in the environment receive tokens from the token manager, and can later use these tokens to access information stored on the original capture device, even by accessing it remotely. Depending on the specific policy regulating environmental capture in a certain place or situation, all mobile devices present in the environment might be informed about the data capture, and provided with the appropriate access tokens. It is also possible to constrain the set of token recipients based on some condition (*e.g.*, that they identify themselves to the token manager).

An alternate configuration would have each client device run its own token manager, which imposes its access policy on the capturing device. In this case, the CA infrastructure on the capture device could require specific conditions to be met on the token managers on client devices, before allowing them to impose tokens on the data. This system would allow later access, and might be more suitable for anonymous access, since the tokens are generated at the client side. Both cases described here assume that the capturing device honors access requests.

If the capture device itself does not have a token manager to introduce its own tokens during capture, a permission-based policy could be envisioned, where the owner of the capture device would need to ask a token holder permit to access the recording. More complex permission policies (*e.g.*, that all people present to a meeting grant permission) would be possible only by changing the core token/condition evaluation algorithm. For example, this could be achieved by adding a different type of tokens that must be all presented in order to gain access, as opposed to the current setup, which requires that only one access token match the stored tokens.

Various other architectures are possible, including: mobile capture and fixed-infrastructure storage and access (useful in open space); fixed-infrastructure capture and mobile storage (useful for exploiting high-quality fixed sensors); and the mobile

device as a control system for environmental data flowing among various fixed-infrastructure components. Although we do not report on these, preliminary analysis suggests that token-based access control could be effectively used for securing these applications as well.

Conclusions and Future Work

We have shown how the unique characteristics and requirements of environmental capture and access systems call for revising our traditional concepts of access control in information systems. The access control system proposed in this article is not intended to be a secure or complete implementation. Instead, we hope that the requirements analysis and sample applications presented here and the rationale for our design decisions can be useful as a base for future work. In particular, the undefined and potentially unlimited user set, the need for flexible policy implementation, and the specific social and perceptual qualities of information sensed from physical environments represent challenges that cannot be solved simply by applying existing models such as RBAC or DRM.

The discussion revealed that the proposed system falls short with relation to several security requirements, including the ability to change access control policies after the fact, the impossibility of limiting token proliferation, and the lack of access control for write and delete operations. However, arguments were provided to point out how it might not be possible to meet some of these requirements even with traditional access control systems and that, in general, these security concerns could be misplaced or even irrelevant to the specific usage contexts and applications of environmental information. Specifically, we highlighted the need for practical security systems able to curb misuse rather than assure high-grade confidentiality.

In our current and future work we are focusing on expanding our understanding of both technical and social needs, norms and rules impacting environmental capture and access security and privacy. We are interested in the role of mobile devices within larger CA systems, both as agents of control by the owners, and as risk factors for third parties. Specifically, we intend to study how social, cognitive and legal needs and constraints impact system design and security choices such access control. We plan to extend the token-based access control model to support CA of audio and video for personal purposes by individuals using mobile, networked devices.

Acknowledgements

<anonymized>

References

1. Abowd, G.D., Classroom 2000: An Experiment with the Instrumentation of a Living Educational Environment, IBM Systems Journal, Special issue on Pervasive Computing, Volume 38, Number 4, pp. 508–530, October 1999.
2. Bellotti, V., Sellen, A., Design for Privacy in Ubiquitous Computing Environments, in Proceedings of European CSCW 93, Milan, Italy
3. Brewer, D., Nash, M.J., The Chinese Wall Security Policy, Proc. IEEE Symposium in Security and Privacy, 1989, Oakland, CA, IEEE Press.
4. Camenish, J., Van Herreweghen, E., Design and Implementation of the idemix Anonymous Credential System, in Proceedings of CCS'02, Nov. 2002, Washington, DC, USA
5. Chaum, D., Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms, Communications of the ACM, vol. 24 no. 2, February, 1981.
6. ContentGuard Holdings, Inc. Extensible Rights Markup Language XrML 2.0 Specification, 2001, http://www.xrml.org
7. Covington, M.J., A Context-Aware Security Architecture for Emerging Applications, in Proceedings of ACSAC '02, Dec. 2002, Las Vegas, Nevada, USA.
8. Deutscher, M., Jeffrey, P., and Siu, N. Information Capture Devices for Social Environments. To appear in Proceedings of Second European Symposium on Ambient Intelligence (EUSAI 2004), November 8-10, Eindhoven, The Netherlands.
9. Duan, Y., Canny, J.: Designing for Privacy in Ubiquitous Computing Environments.
10. Graham, G.S., Denning, P.J., Protection – Principles and practice, Proceedings of AFIPS 1972 Joint Spring Computer Conference, Vol. 40, AFIPS Press, pp. 417–429.
11. Harrison, S., Dourish, P., Re-place-ing space: The roles of space and place in collaborative systems. In: Proceedings of the 1996 ACM Conference on Computer Supported Cooperative Work (CSCW'96). ACM Press, New York (1996) 67–76
12. Hayes G., Patel, S.N., Truong, K.N., Iachello, G., Kientz, J.A., Farmer, R., Abowd, G.D. The Personal Audio Loop: Designing a Ubiquitous Audio-Based Memory Aid, Proc. Mobile HCI 2004, LNCS 3160, Springer Verlag, 168–179.
13. Langheinrich, M., Coroama, v., Bohn, J., Rohs, M. As we may live – Real-world implications of ubiquitous computing, available at http://www.vs.inf.ethz.ch/publ/papers/uc-implications.pdf
14. Macedo, A. A., Truong, K. N., Camacho-Guerrero, J. A., Pimentel, M. d. G.: Automatically Sharing Web Experiences through a Hyperdocument Recommender System. In: Proc. Conference on Hypertext and Hypermedia (2003) 48 – 56
15. Moran, T. P., Palen, L., Harrison, S., Chiu, P., Kimber, D., Minneman, S., van Melle, W., and Zellweger, P., "I'll get that off the audio": A case study of salvaging multimedia meeting records. Proceedings of the CHI'97 Conference on Human Factors in Computer Systems, 1997, New York: ACM.
16. Organization for Economic Cooperation and Development, Guidelines on the Protection of Privacy and Transborder Flows of Personal Data, 1980, http://www.oecd.org
17. Palen, L., Dourish, P., Unpacking "Privacy" for a Networked World, in Proceedings of CHI2003, ACM Press
18. Povey, D., Optimistic Security: A New Access Control Paradigm, Proc. 1999 New Security Paradigms Workshop, Ontario, Canada, ACM Press, pp 40–45.
19. Truong, K.N., Abowd, G.D., INCA: A Software Infrastructure to Facilitate the Construction and Evolution of Ubiquitous Capture & Access Applications, in Proceedings of Pervasive 2004: pp. 140–157

20. Sampemane, G., et al., Access Control for Access Spaces, in Proceedings of ACSAC '02, Dec. 2002, Las Vegas, Nevada, USA.
21. Sumi, y., Ito, S., Matsuguchi, t., Fels, S., Mase, K.: Collaborative Capturing and Interpretation of Interactions, proceedings of Pervasive 2004 Workshop on Memory and Sharing of Experiences, April 20, 2004, Vienna, Austria. http://www.ii.ist.i.kyoto-u.ac.jp/~sumi/pervasive04/
22. Weiser, M., Some Computer Science Problems in Ubiquitous Computing, Communications of the ACM, July 1993.
23. White, D.R., Camacho-Guerrero, J.R., Truong, K.N., Abowd, G.D., Morrier, M.J., Vekaria, P.C., Gromala, D. Mobile Capture and Access for Assessing Language and Social Development in Children with Autism. In the Extended Abstracts of UBICOMP 2003 (October 12–15, Seattle, WA), 2003, pp. 137–140.